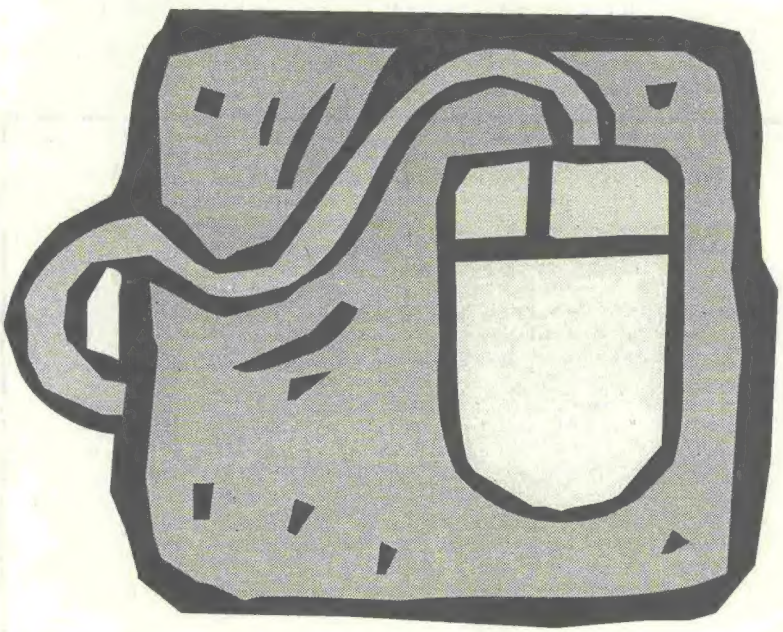
ณรงค์ จันทโพธิ์

สร้างสรรค์การใช้งานเม้าส์ด้วย OOP : สร้างกรอบข้อความ



แนะนำการสร้างออบเจ็กต์ กรอบข้อความ 2 แบบ

สวัสดีครับท่านผู้อ่านพบกันอีกแล้วครับ จากตอนที่ผ่านมาทั้ง 3 ตอนนั้นออบเจ็กต์ที่สร้างกันไว้ก็พอที่จะนำมาสร้างโปรแกรมง่าย ๆ กันบ้างแล้ว แต่ถ้าเพื่อจะให้สมบูรณ์ยิ่งขึ้น ในตอนนี้ผมจะแนะนำการสร้างออบเจ็กต์กรอบข้อความ สำหรับกรอบข้อความที่จะออกแบบนี้ผมได้แบ่งออกเป็น 2 แบบ คือ

1. แบบเก็บพื้นหลังก่อนแสดง กรอบข้อความแบบนี้จะมีการเก็บพื้นที่ของจอในบริเวณที่จะแสดงกรอบข้อความ และเมื่อสิ้นสุดการทำงานของกรอบข้อความก็จะคืนพื้นที่ส่วนดังกล่าว

2. แบบไม่เก็บพื้นหลัง กรอบข้อความแบบนี้หลังจากทำงานเสร็จแล้วจะต้องทำการวาดส่วนพื้นหลังใหม่

ลักษณะของกรอบข้อความที่จะทำการออกแบบนี้จะมีส่วนประกอบดังนี้ครับ

- เป็นกรอบแบบ 3 มิติ
- มีส่วนของการแสดงชื่อของกรอบข้อความ (Title bar)
- มีปุ่มเปลี่ยนภาษาไทย/อังกฤษ (สามารถกำหนดให้มีหรือไม่มีก็ได้)
- มีปุ่มปิดกรอบข้อความ (สามารถกำหนดให้มีหรือไม่มีก็ได้)

เริ่มออกแบบ

ต่อไปเราก็จะออกแบบกันเลยครับ เริ่มต้นมาพิจารณาที่ความสามารถในการเก็บพื้นหลังกันก่อนโดยใช้ฟังก์ชันกราฟิกที่มีมากับ BorlandC คือฟังก์ชัน

```
Getimage(int SCol,int SRow,int
ECol,int ERow,void far *Image)
```

แต่เนื่องจากตัวแปร Image นั้นเป็นพื้นที่หน่วยความจำที่จะใช้ในการเก็บพื้นหลัง โดยต้องมีขนาดพอที่จะเก็บพื้นหลังนั้นได้ ในการจะจองพื้นที่ให้มากพอที่จะเก็บได้ก็จะต้องรู้ขนาดของพื้นที่โดยใช้ฟังก์ชัน

```
imagesize(int SCol,int SRow,int
ECol,int ERow)
```

ในการใช้งานจะกระทำได้ดังนี้

```
Image = new char[imagesize
(SCol,SRow,ECol,HalfPoint)]
```

แต่เนื่องจากขนาดของหน่วยความจำที่จะใช้ในการเก็บภาพนั้นมีขนาดสูงสุดคือ 64 กิโลไบต์ แต่การเก็บภาพทั้งหน้าจอจะใช้หน่วยความจำมากกว่า 64 กิโลไบต์ ดังนั้นจึงต้องใช้ตัวแปร 2 ตัว โดยจะแบ่งเก็บตัวละครึ่งภาพในการหาตำแหน่งกลางก็ใช้การคำนวณธรรมดาดังนี้

```
HalfPoint = (SRow+ERow)/2
```

สำหรับการคืนพื้นที่หน้าจอหลังจากการทำงานของกรอบข้อความนั้นเสร็จสิ้นก็จะใช้ฟังก์ชันที่มีมากับ BorlandC คือ

```
putimage(int Col,int Row,void far
*Image,int operation)
```

Col, Row เป็นพิกัดที่จะทำการวาดภาพกลับลงไป

Image เป็นหน่วยความจำที่เก็บภาพไว้

operation เป็นรูปแบบที่จะทำการวาดภาพซึ่งมี 5 รูปแบบ คือ

- COPY_PUT เป็นการวางรูปทับกับรูปบนหน้าจอ
- XOR_PUT ภาพที่นำไปวางจะ XOR กับภาพบนหน้าจอ
- OR_PUT ภาพที่นำไปวางจะ OR กับภาพบนหน้าจอ
- AND_PUT ภาพที่นำไปวางจะ AND กับภาพบนหน้าจอ
- NOT_PUT ภาพที่นำไปวางจะ Inverse กับภาพบนหน้าจอ

ในรูปแบบ XOR, OR, AND และ NOT นั้น ปรากฏการณ์ดังกล่าวจะทำให้สีของภาพที่นำไปวางจะเปลี่ยนไป

สำหรับกระบวนการทำงานที่สมบูรณ์อยู่ในวิธีการ SaveImage() และ ReturnImage() ตั้งแต่บรรทัดที่ 77 ถึง 95 ในลิสติงที่ 1 ครับ

ลิสติงที่ 1 WINDOW.CPP

```
1.    #include"objects.h"
2.    #include"mouobj.h"
3.    #include"stdfon.h"
4.    int Thai=0;
5.    int LangKey = ``';
6.    extern int MouX,MouY;
7.    extern WORD MouEvent;
8.    //*************************************************************************
9.    void DeepRectangle(int X1,int Y1,int X2,int Y2,BYTE Style){
10.   int UpColor;
11.   int DwnColor;
12.   if(Style == 0){
13.   UpColor  = DARKGRAY;
14.   DwnColor = WHITE;
15.   }
16.   else{
17.   UpColor  = WHITE;
18.   DwnColor = DARKGRAY;
19.   }
20.   HideMouse();
21.   setcolor(UpColor);
22.   rectangle(X1,Y1,X2,Y2);
23.   setcolor(DwnColor);
24.   line(X1+1,Y2,X2,Y2);
25.   line(X2,Y1+1,X2,Y2);
26.   ShowMouse();
27.   }
28.   //*************************************************************************
29.   void ThreeDRectangle(int X1,int Y1,int X2,int Y2){
30.   HideMouse();
31.   setcolor(DARKGRAY);
32.   rectangle(X1,Y1,X2-1,Y2-1);
33.   setcolor(WHITE);
34.   rectangle(X1+1,Y1+1,X2,Y2);
35.   ShowMouse();
36.   }
37.   //*************************************************************************
38.   void GrWindow(int Color,int Col,int Row,int Width,int Height,int TitleBColor){
39.   const FSize = 5;
40.   HideMouse();
41.   setfillstyle(1,Color);
42.   bar(Col,Row,Col+Width,Row+Height);
43.   if(TitleBColor!=NULL){
44.   DeepRectangle(Col+FSize,Row+FSize,Col+Width-FSize,Row+27,0);
45.   ThreeDRectangle(Col+FSize,Row+FSize+27,Col+Width-FSize,Row+Height-FSize);
46.   setfillstyle(1,TitleBColor);
47.   bar(Col+FSize+1,Row+FSize+1,Col+Width-FSize-1,Row+27-1);
48.   }
49.   else
50.   DeepRectangle(Col+FSize,Row+FSize,Col+Width-FSize,Row+Height-FSize,0);
51.   DeepRectangle(Col,Row,Col+Width,Row+Height,1);
52.   ShowMouse();
53.   }
```

→

ลิสติงที่ 1 (ต่อ)

```
54.  /*************************************************************************/
55.  Window::Window(char *Txt,int X,int Y,int Width,int High,WORD style,
56.  int TxtColor,int TtlColor,int WColor){
57.  SCol     = X;
58.  SRow     = Y;
59.  ECol     = X+Width;
60.  ERow     = Y+High;
61.  Style    = style;
62.  ObjType  = WINDOW;
63.  RetCmd   = CLOSEWIN;
64.  Color    = TxtColor;
65.  WinColor= WColor;
66.  TBarColor = TtlColor;
67.  Cmd       = LangKey;
68.  Title     = new char[strlen(Txt)+1];
69.  strcpy(Title,Txt);
70.  if(style&FOCUSED)
71.  style^=FOCUSED;
72.  if(Style&CAPTION)
73.  SaveImage();
74.  Draw();
75.  }
76.  //************************************************************************
77.  void Window::SaveImage(){
78.  int HalfPoint = (SRow+ERow)/2;
79.  Image1 = new char[imagesize(SCol,SRow,ECol,HalfPoint)];
80.  Image2 = new char[imagesize(SCol,HalfPoint+1,ECol,ERow)];
81.  HideMouse();
82.  getimage(SCol,SRow,ECol,HalfPoint,Image1);     //Save first block
83.  getimage(SCol,HalfPoint+1,ECol,ERow,Image2);  //Save second block
84.  ShowMouse();
85.  }
86.  //************************************************************************
87.  void Window::ReturnImage(){
88.  int h = (SRow+ERow)/2;
89.  HideMouse();
90.  putimage(SCol,SRow,Image1,COPY_PUT);          //Display first block
91.  putimage(SCol,h+1,Image2,COPY_PUT);          //Display second block
92.  ShowMouse();
93.  delete Image1;
94.  delete Image2;
95.  }
96.  //************************************************************************
97.  void Window::Destroyed(){
98.  if(Style&CAPTION)
99.  ReturnImage();
100. delete(Title);
101. }
102. //************************************************************************
103. void Window::ShowLang(int Up){
104. char *L = "E";
105. HideMouse();
106. if(Thai)
107. = '·';
108. else
109. = 'E';
110. if(Up)
111. OutTextXY(SCol+21,SRow+10,L,BLACK);
112. else
113. OutTextXY(SCol+22,SRow+11,L,BLACK);
114. ShowMouse();
115. }
116. //************************************************************************
117. void Window::WinButtDwn(int Lang){
118. int x0,x1,y0=SRow+7,y1=SRow+25;
119. if(Lang){
120. x0 = SCol+7;
121. x1 = SCol+29;
122. }
123. else{
124. x0 = ECol-25;
125. x1 = ECol-7;
```

→

ลิสติงที่ 1 (ต่อ)

```
126.  }
127.  HideMouse();
128.  setfillstyle(1,LIGHTGRAY);
129.  bar(x0,y0,x1,y1);
130.  setcolor(DARKGRAY);
131.  line(x0,y0,x1,y0);   //Hor line
132.  line(x0,y0+1,x0,y1); //Ver line
133.  ShowMouse();
134.  }
135.  //*************************************************************************
136.  void Window::WinButtUp(int Lang){
137.  int x0,x1,y0=SRow+7,y1=SRow+25,I;
138.  if(Lang){
139.  x0 = SCol+7;
140.  x1 = SCol+29;
141.  }
142.  else{
143.  x0 = ECol-25;
144.  x1 = ECol-7;
145.  }
146.  HideMouse();
147.  setfillstyle(1,LIGHTGRAY);
148.  bar(x0,y0,x1,y1);
149.  setcolor(BLACK);
150.  rectangle(x0-1,y0-1,x1+1,y1+1); //Top line
151.  setcolor(WHITE);
152.  for(I=0;I<2;I++){
153.  line(x0,y0+I,x1-I,y0+I);      //Hor line
154.  line(x0+I,y0+1,x0+I,y1-I);    //Ver line
155.  }
156.  setcolor(DARKGRAY);
157.  for(I=0;I<2;I++){
158.  line(x0+I,y1-I,x1,y1-I);      //Hor  line
159.  line(x1-I,y0+I,x1-I,y1);      //Ver line
160.  }
161.  ShowMouse();
162.  }
163.  //*************************************************************************
164.  void Window::DrawLangButtDwn(){
165.  unsigned char *Pic;
166.  WinButtDwn(1);
167.  ShowLang(0);  //Down
168.  }
169.  //*************************************************************************
170.  void Window::DrawLangButtUp(){
171.  WinButtUp(1);
172.  ShowLang(1);  //Up
173.  }
174.  //*************************************************************************
175.  void Window::DrawCloseButtUp(){
176.  WinButtUp(0);
177.  HideMouse();
178.  setcolor(BLACK);
179.  circle(ECol-16,SRow+16,4);
180.  line(ECol-16,SRow+11,ECol-16,SRow+16);
181.  setcolor(LIGHTGRAY);
182.  line(ECol-15,SRow+11,ECol-15,SRow+15);
183.  line(ECol-17,SRow+11,ECol-17,SRow+15);
184.  ShowMouse();
185.  }
186.  //*************************************************************************
187.  void Window::DrawCloseButtDwn(){
188.  WinButtDwn(0);
189.  HideMouse();
190.  setfillstyle(1,LIGHTGRAY);
191.  bar(ECol-23,SRow+9,ECol-9,SRow+23);
192.  setcolor(BLACK);
193.  circle(ECol-15,SRow+17,4);
194.  line(ECol-15,SRow+12,ECol-15,SRow+17);
195.  setcolor(LIGHTGRAY);
196.  line(ECol-14,SRow+11,ECol-14,SRow+15);
```

→

ลิสติงที่ 1 (ต่อ)

```
197.  line(ECol-16,$Row+11,ECol-16,SRow+15);
198.  ShowMouse();
199.  }
200.  //*****************************************************************************
201.  void Window::Draw(){
202.  int Shift =22*((Style&LANG_KEY)>>3) + (18*((Style&SYS_CLOSE)>>1));
203.  //22 is Lang key button 's size , 18 is Sys close button 's size
204.  Style |= VISIBLE;
205.  GrWindow(WinColor,SCol,SRow,ECol-SCol,ERow-SRow,TBarColor);
206.  HideMouse();
207.  if(Style&SYS_CLOSE)
208.  DrawCloseButtUp();
209.  if(Style&LANG_KEY)
210.  DrawLangButtUp();
211.  ShowText(SCol+(22*((Style&LANG_KEY)>>3)),SRow,(ECol-SCol)-Shift,32,Title,Color);
212.  ShowMouse();
213.  }
214.  //*****************************************************************************
215.  void Window::SetLanguage(){
216.  Thai ^= 1;
217.  DrawLangButtUp();//Up
218.  }
219.  //*****************************************************************************
220.  int Window::GetButtRelease(){
221.  int Close=0,Active=3;
222.  if((MouEvent==L_DOWN)&&MouInBox(MouX,MouY,ECol-25,SRow+6,ECol-6,SRow+26)){
223.  Close = 1;
224.  DrawCloseButtDwn();
225.  }
226.  else
227.  if((MouEvent==L_DOWN)&&MouInBox(MouX,MouY,SCol+6,SRow+6,SCol+30,SRow+26))
228.  DrawLangButtDwn();
229.  while(Active==3){
230.  GetMouInfo(&MouEvent,&MouX,&MouY);
231.  if(MouEvent==NO_DOWN){
232.  if(MouInBox(MouX,MouY,ECol-25,SRow+6,ECol-6,SRow+26))//Close
233.  Active = 1;
234.  else
235.  if(MouInBox(MouX,MouY,SCol+6,SRow+6,SCol+30,SRow+26))
236.  Active = 2;
237.  }
238.  else
239.  if((MouEvent==L_DOWN)&&
240.  (!MouInBox(MouX,MouY,ECol-25,SRow+6,ECol-6,SRow+26))&&
241.  (!MouInBox(MouX,MouY,SCol+6,SRow+6,SCol+30,SRow+26)))
242.  Active = 0;
243.  }
244.  if(Active==2){
245.  SetLanguage();
246.  Active = 0;
247.  }
248.  if(Close)
249.  DrawCloseButtUp();
250.  else
251.  DrawLangButtUp();
252.  return Active;
253.  }
254.  //*****************************************************************************
255.  int Window::Pressed(){
256.  if((MouEvent==L_DOWN)&&MouInBox(MouX,MouY,ECol-25,SRow+6,ECol-6,SRow+26)&&
257.  (Style&SYS_CLOSE))
258.  return CLOSEWIN;
259.  if((MouEvent==L_DOWN)&&MouInBox(MouX,MouY,SCol+6,SRow+6,ECol-6,SRow+26)&&
260.  (Style&LANG_KEY))
261.  return 1;
262.  return 0;
263.  }
264.  //*****************************************************************************
```

■

จากลิสติงที่ 1 นี้มีฟังก์ชันที่สามารถใช้งานทั่วไปได้ 2 ฟังก์ชัน คือ

```
-void DeepRectangle(int X1,int Y1,int X2,
int Y2,BYTE Style)
```

เป็นฟังก์ชันที่ทำหน้าที่วาดกรอบสี่เหลี่ยมบนหน้าจอ โดยมีรายละเอียดของพารามิเตอร์ดังนี้

X1,Y1	เป็นพิกัดเริ่มต้นของกรอบ
X2,Y2	เป็นพิกัดสิ้นสุดของกรอบ
Style	เป็นรูปแบบของกรอบสี่เหลี่ยมโดยมีค่าเป็น
0	กรอบมีลักษณะเป็นกรอบลึก
1	กรอบมีลักษณะเป็นกรอบลอย

```
-void ThreeDRectangle(int X1,int Y1,int
X2,int Y2)
```

เป็นฟังก์ชันที่ทำหน้าที่วาดกรอบสี่เหลี่ยมแบบ 3 มิติบนหน้าจอ โดยมีรายละเอียดของพารามิเตอร์ดังนี้

X1,Y1	เป็นพิกัดเริ่มต้นของกรอบ
X2,Y2	เป็นพิกัดสิ้นสุดของกรอบ

ส่วนรายละเอียดการใช้งานของ ออบเจ็กต์ Window มีดังนี้

```
Window(char *Txt,int X,int Y,int Width,int
Height,WORD style,int TxtColor,int TitleBCOlor,
int Wcolor)
```

พารามิเตอร์ :

Txt	เป็นข้อความที่แสดงบน Title bar
X,Y	เป็นพิกัดเริ่มต้นของกรอบ
Width	เป็นความกว้างของกรอบ
Height	เป็นความสูงของกรอบ
style	เป็นรูปแบบของกรอบข้อความ โดยรูปแบบนี้ประกอบด้วย
	VISIBLE ให้มีการวาดรูปทันทีหลังจากประกาศ ออบเจ็กต์
	CAPTION ให้มีการเก็บพื้นหลังก่อนแสดง
	LANG_KEY ให้มีปุ่มเปลี่ยนภาษา
	SYS_CLOSE ให้มีปุ่มปิดกรอบข้อความ

จะสังเกตเห็นว่าออบเจ็กต์ Window นี้ไม่มีการกำหนด Message ส่งกลับ แต่เมื่อพิจารณาภายใน constructor (ลิสติงที่ 1 บรรทัดที่ 63) มีการกำหนด

```
RetCmd = CLOSEWIN
```

ซึ่งเป็นการกำหนดไว้ตายตัวให้กับ Message ส่งกลับของ Window เมื่อมีการกดปุ่มปิดกรอบ นอกจากนี้บรรทัดที่มีการกำหนดค่า RefKey (Cmd) ไว้ดังนี้

```
Cmd = LangKey
```

หมายถึงมีการกำหนดค่าสแกนโค้ดของ Keyboard ให้มีค่าเดียวกันกับค่าสแกนโค้ดของ Keyboard ที่ใช้ในการสลับระหว่าง ภาษาไทย/อังกฤษ ผมได้กำหนดให้เป็น key "~" (tide) เป็น key สลับระหว่างภาษาไทย/อังกฤษ (บรรทัดที่ 5 ในลิสติงที่ 1)

สำหรับการทำงานของ Window เมื่อมีการถูกกด (ด้วยเม้าส์) นั้น จะแตกต่างจากออบเจ็กต์ที่ผ่านมาคือ จะต้องมีการตรวจสอบการถูกกดถึง 2 แห่ง คือ ส่วนของปุ่มกดเปลี่ยนภาษา และส่วนของปุ่มกดปิดกรอบข้อความพิจารณาได้ในลิสติงที่ 1 บรรทัดที่ 256-258 เป็นการตรวจสอบการถูกกดของปุ่มกดปิดกรอบข้อความ หากถูกกดจะ return Message CLOSEWIN ออกมาเพื่อบอกให้โปรแกรมรู้ ส่วนบรรทัดที่ 259-261 เป็นการตรวจสอบการถูกกดของปุ่มกดปิดกรอบข้อความ หากถูกกดจะ return ค่า 1 ออกมาซึ่ง Message ทั้งสองนี้ จะถูกนำไปตรวจสอบใน Method GetButtRelease() อีกทีหนึ่งซึ่ง Message ที่จะถูกส่งออกไปจริงมีเพียง Message CLOSEWIN ส่วน Message จากการกดปุ่มเปลี่ยนภาษานั้นจะถูกใช้ในการทำงานภายในออบเจ็กต์เท่านั้น

การตรวจจับ Message ให้กับ Window

ในส่วนของ Message loop ที่ต้องคอยดักเหตุการณ์ของคารกดคีย์บอร์ดนั้นต้องเพิ่มการตรวจสอบการกดคีย์บอร์ดใน key ที่กำหนดให้เป็น key เปลี่ยนภาษา ในขั้นตอนการตรวจสอบเป็นดังลิสติงที่ 2

ลิสติงที่ 2

```
1: if((Msg&0x00FF)==LangKey){
2:  while(CurrObj!=NULL){
3:    if((CurrObj-
>ObjType==WINDOW)&&(CurrObj->Style&LANG_KEY)){
4:    CurrObj->SetLanguage();
5:    return ID_UNDEF;
6:    }
7:    if(CurrObj==NULL) //no Window
8:    Thai ^= 1;  //set system language
9:         return ID_UNDEF;
10: }
```

การปรับปรุง Code เก่า

บทความในตอนนี้มีการเพิ่มออบเจ็กต์ Window และมีการเพิ่มส่วนการทำงานในการตรวจสอบการกดคีย์บอร์ด ซึ่งมีรายละเอียดดังนี้

1. OBJINTER.CPP

- เพิ่มการประกาศตัวแปรที่ส่วนหัวของ file ดังนี้

```
extern WORD LangKey;
extern int Thai;
```

- ใน Method GetMessage() จาก

```
switch(Msg){
        .
        .
        .
        }
CurrObj = FirstObj;
```

ให้นำลิสติงที่ 2 มาต่อครับ

2. OBJECTS.H

- เพิ่มการประกาศส่วนของการ define ดังนี้

```
#define WINDOW        10
#define CLOSEWIN      12
const WORD CAPTION    = 32;

const WORD SYS_CLOSE = 2;
const WORD LANG_KEY  = 8;
```

- เพิ่มการประกาศ Method ใน class AppObj ดังนี้

```
virtual void SetLanguage(){};
```

- เพิ่มการประกาศออบเจ็กต์ Window ดังนี้

```
1: class Window : public AppObj{
2: //private:
3:    int  TBarColor,WinColor;
4:    char *Image1,*Image2;
5:    void WinButtUp(int);
6:    void WinButtDwn(int);
7:    void DrawCloseButtUp();
8:    void DrawCloseButtDwn();
9:    void DrawLangButtUp();
10:   void DrawLangButtDwn();
11:   void ShowLang(int);
12:   void SaveImage();
13:   void ReturnImage();
14: public:
15: Window(char *Txt,int X,int Y,int
      Width,int High,WORD style,
16: int TxtColor,int TtlColor,int
      WinColor);
17: protected:
18:   void Destroyed();
19:   int  GetButtRelease();
20:   void SetLanguage();
21:   int  Pressed();
22:   void Draw();
23: };
```

ทดลองใช้งาน

เช่นเคยครับ เมื่อทำการสร้างออบเจ็กต์ใหม่พร้อมทั้งปรับปรุงการทำงานบางส่วนดังที่กล่าวมาแล้วเราก็ทดลองใช้งานดู ดังโปรแกรมตัวอย่างจากลิสติงที่ 3

ลิสติงที่ 3 SAMPLE.CPP

```
1.  #include<graphics.h>
2.  #include"objects.h"
3.  #include"objinter.h"
4.  #include"mouobj.h"
5.  #include"stdfon.h"
6.  #define ID_QUIT 10
7.  #define ID_TEST 101
8.  int     MouX,MouY;
9.  WORD    MouEvent,AppMsg;
10. GUIApp  TestApp;
11. int     Quit = 0;
12. //*************************************************************************
13. void TestWindow(){
14. GUIApp TestW;
15. WORD   TestMsg=0;
16. HideMouse();
17. TestW.AddObj(new Window("Test PopUp Window",140,80,350,250,VISIBLE|SYS_CLOSE|
18. LANG_KEY|CAPTION,WHITE,BROWN,LIGHTGRAY));
19. TestW.AddObj(new TextButton(282,290,60,25,"Close",VISIBLE|FOCUSED,27,CLOSEWIN));
20. line(160,110,180,150);
21. line(180,150,303,150);
22. OutTextXY(190,135,"Language button",RED);
23. setcolor(BLACK);
24. line(475,110,435,180);
25. line(335,180,435,180);
26. OutTextXY(343,165,"Close button",BLUE);
27. ShowMouse();
```

ลิสติงที่ 3 (ต่อ)

```
28.   while(TestMsg!=CLOSEWIN){          //Wait for CLOSEWIN Message
29.   GetKbdMouse(&MouEvent,&MouX,&MouY);
30.   TestMsg = TestW.GetMessage(MouEvent,MouX,MouY);
31.   }
32.   }
33.   //*************************************************************************
34.   void DoMessage(WORD *Msg){
35.   switch(*Msg){
36.   case CLOSEWIN :
37.   case ID_QUIT  :
38.   case 0x2d00   : Quit = 1;       //Alt-X key
39.   break;
40.   case ID_TEST  : TestWindow();
41.   break;
42.   }
43.   }
44.   //*************************************************************************
45.   main(){
46.   int gd= DETECT,gm;
47.   initgraph(&gd,&gm,"c:\\lang\\borlandc\\bgi");
48.   TestApp.AddObj(new Window("OOP Programming 4 : Create Window",0,0,639,479,VISIBLE|
49.   SYS_CLOSE,WHITE,BLUE,LIGHTGRAY));
50.   TestApp.AddObj(new TextButton(40,50,120,25,"PopWindow",VISIBLE|FOCUSED,NOKEY,ID_TEST));
51.   OutThaiXY(130,150,"·´สอบการสร้างกรอบข้อความ ด้วยการเขียนโปรแกรมแบบ OOP ",GREEN);
52.   OutThaiXY(260,230,"โดย ณรงค์ จันทโพธิ์",MAGENTA);
53.   ShowMouse();
54.   while(!Quit){
55.   GetKbdMouse(&MouEvent,&MouX,&MouY);
56.   AppMsg = TestApp.GetMessage(MouEvent,MouX,MouY);
57.   DoMessage(&AppMsg);
58.   }
59.   HideMouse();
60.   closegraph();
61.   return 0;
62.   }
```

โดยมีรายละเอียดการทำงานของโปรแกรมดังนี้

บรรทัดที่ 1-5 เป็นการประกาศ header file ที่ต้องใช้

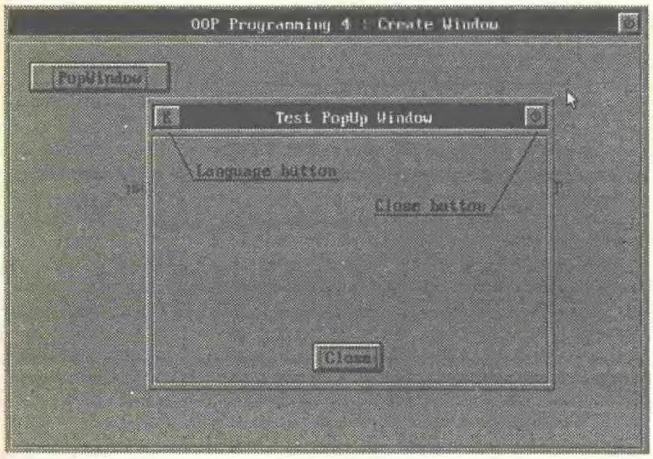
บรรทัดที่ 6-7 เป็นการประกาศ Message ที่จะใช้ในโปรแกรม

บรรทัดที่ 8-11 เป็นการประกาศ ตัวแปรที่ต้องใช้

บรรทัดที่ 13-32 เป็นฟังก์ชันที่ใช้ทดสอบการทำงานของกรอบข้อความ ส่วนสำคัญอยู่ที่การประกาศสร้างออบเจ็กต์ Window ที่มีรูปแบบที่เก็บพื้นหลังก่อน (CAPTION) ในฟังก์ชันนี้มีการประกาศ instance ของ GUIApp ขึ้นมาใหม่คือ TestW ดังบรรทัดที่ 14 ผลจากการประกาศเช่นนี้เมื่อสิ้นสุดการทำงานของฟังก์ชัน TestWindow() จะเกิดการทำงานของ Destructor ของ GUIApp โดยทันที

บรรทัดที่ 34-43 เป็นฟังก์ชัน Domessage() ซึ่งทำงานตาม Message ที่เกิดจากโปรแกรมหลัก

บรรทัดที่ 48 เป็นการประกาศสร้างออบเจ็กต์ Window ที่ไม่เก็บพื้นหลังก่อน โดยปกติแล้วกรอบข้อความของโปรแกรมหลักไม่ควรจะเป็นแบบเก็บพื้นหลัง เนื่องจากว่าเมื่อจบการทำงานของกรอบข้อความหลักของโปรแกรมก็ถือว่าเป็นการจบการทำงานของโปรแกรมนั่นเอง (ดูรูปที่ 1)



รูปที่ 1 ผลจากการทำงานของโปรแกรม

ส่งท้าย

เนื่องจากการเขียนโปรแกรมแบบ OOP นี้จะมีผลทำให้ขนาดของโปรแกรมใหญ่ขึ้น ซึ่งขนาดของ heap จะเหลือน้อยลง ในการประกาศออบเจ็กต์ Window ที่มี style แบบ CAPTION ต้องระมัดระวังในการกำหนดขนาดของกรอบข้อความ เนื่องจากถ้ากำหนดขนาดใหญ่

เกินขนาดของ heap ที่เหลืออยู่จะมีผลทำให้โปรแกรมทำงานผิดพลาดได้ ซึ่งขนาดในโปรแกรมตัวอย่างที่ผมให้มานี้เป็นกรอบข้อความที่มีขนาดที่โตที่สุดเท่าที่จะกำหนดได้แล้ว โดยปกติเราจะต้องทำการคอยตรวจสอบหน่วยความจำที่ขอจาก heap ว่าการขอนั้นสำเร็จหรือไม่ ถ้าไม่สำเร็จควรจะมีข้อความเตือนและไม่ทำงานในส่วนนั้นต่อไป ในการตรวจสอบส่วนนี้ผมไม่ได้เขียนไว้ด้วย ก็ขอยกให้เป็นหน้าที่ของท่านผู้อ่านก็แล้วกันครับ โดยผมจะยกตัวอย่างให้พิจารณาดังนี้

```
        Image = new char[65535];
        if(Image==NULL){
              printf("Not enough memory to
allocate\n");
        return;
        }
```

ก่อนจบบทความ ผมขอสรุปถึงไฟล์ที่ต้องใช้ในการคอมพายล์ ซึ่งมีรายละเอียดดังนี้ครับ

```
1.OBJECTS.H
2.OBJINTER.H
3.MOUOBJ.H
4.STDFON.H
5.THAIENG1.TXT
6.STDFON.CPP
7.MOUOBJ.CPP
8.RADIOBTN.CPP
9.CHECKBOX.CPP
10.BUTTON.CPP
11.WINDOW.CPP
12.OBJINTER.CPP
13.SAMPLE.CPP
```

โดยที่รายการ 1, 2, 12 เป็นไฟล์เก่าจากตอนที่ 3 ที่ได้ปรับปรุงตามเนื้อหาในตอนนี้

รายการ 11, 13 เป็นไฟล์ใหม่ จากเนื้อหาในตอนนี้

รายการ 3, 4, 5, 6, 7, 8, 9, 10 เป็นไฟล์เก่าจากตอนที่ 3

ไฟล์ทั้งหมดนี้มีให้บริการในซอฟต์เมล์เช่นเคยครับ และสำหรับท่านผู้อ่านท่านใดมีปัญหาหรือคำแนะนำประการใดก็สามารถติดต่อผมทาง E-mail ตามที่อยู่ดังนี้ครับ

narong@ram1.ru.ac.th

Kaset Mini-MIS
มหาวิทยาลัยเกษตรศาสตร์
โดยภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์

เปิดโครงการฝึกอบรมผู้บริการในหลักสูตร ระบบสารสนเทศเพื่อการจัดการ (Manager Information Systems) **Kaset Mini-MIS** โดยหลักสูตรมีองค์ประกอบดังนี้คือ การบริหารโครงการด้วยระบบสารสนเทศ เทคโนโลยีคอมพิวเตอร์ ฮาร์ดแวร์ และซอฟต์แวร์ เทคโนโลยีระบบฐานข้อมูล อินเทอร์เน็ต/อินทราเน็ต EDI และ Electronic Commerce ระบบความปลอดภัยของคอมพิวเตอร์ เทคโนโลยีมัลติมีเดียและWorld Wide Web การสื่อสารข้อมูลและเครือข่ายคอมพิวเตอร์ สำนักงานอัตโนมัติและซอฟต์แวร์สนับสนุน

โดยวิธีการอบรม/สัมมนา จะจัดให้มีการเรียนการสอนในลักษณะการบรรยายในชั้นเรียน และการลงมือปฏิบัติในห้องทดลองด้วยระบบคอมพิวเตอร์ที่ทันสมัย พร้อมทั้งมีการสัมมนาดูงานทางด้านเทคโนโลยีคอมพิวเตอร์ที่ประเทศเยอรมันนี และการดูงานในประเทศ

ระยะเวลาสัมมนา
ตั้งแต่วัน 14 กุมภาพันธ์
ถึง 14 มิถุนายน 2540
ประมาณ 200 ชั่วโมง

กำหนดขายใบสมัคร
2 มกราคม - 31 มกราคม 2540
ณ ตึกที่ทำการภาควิชาคอมพิวเตอร์ ชั้น 3
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์

กำหนดรับสมัครและสอบสัมภาษณ์
27-31 มกราคม 2540

ค่าธรรมเนียมตลอดหลักสูตร
90,000 บาท
(รวมค่าเอกสาร/การเดินทางไปดูงานในและต่างประเทศ)

สอบถามรายละเอียดได้ที่
โทร. 661-4621-25 ต่อ 191,183
แฟกซ์ 579-6245